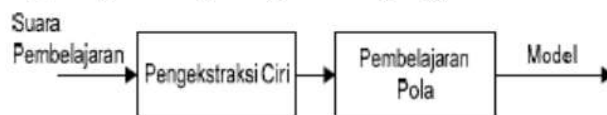


Jaringan Syaraf Tiruan pada Robot

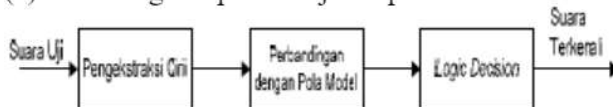
Membuat aplikasi pengenalan suara untuk pengendalian robot dengan menggunakan jaringan syaraf tiruan sebagai algoritma pembelajaran dan pemodelan dalam pengenalan suara.

Pengenalan Suara

Salah satu bentuk pendekatan pengenalan suara adalah pendekatan pengenalan pola yang terdiri dari dua langkah yaitu pembelajaran pola suara dan pengenalan suara melalui perbandingan pola. Tahap perbandingan pola adalah tahap saat suara yang akan dikenali dibandingkan polanya dengan setiap kemungkinan pola yang telah dipelajari dalam fase pembelajaran, untuk kemudian diklasifikasikan dengan pola terbaik yang cocok. Blok diagram pembelajaran pola dan pengenalan suara ditunjukkan pada gambar di bawah ini.



(a) Blok diagram pembelajaran pola.

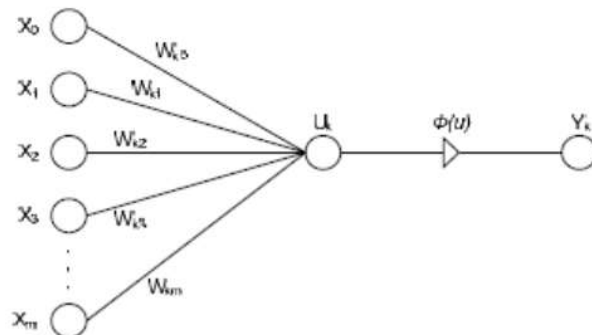


(b) Blok diagram pengenalan suara.

Pengenalan suara secara umum dapat dibagi menjadi tiga tahap, yaitu tahap ekstraksi ciri, tahap pemodelan atau pembelajaran, dan tahap pengenalan suara.

Pemodelan Jaringan Syaraf Tiruan

Model matematis dari neuron dapat dilihat pada Gambar. Masukan dan bobot merupakan model matematis dari dendrit, penjumlahan serta fungsi pengaktif merupakan model matematis dari soma (badan sel), serta keluaran merupakan model matematis dari akson.



Model matematis neuron

Setiap jalur nilai masukan ke neuron mempunyai bobot, dan terdapat satu jalur masukan tambahan yang bukan merupakan jalur masukan, yaitu jalur ke-0. Jalur ke-0 ini dapat berfungsi sebagai nilai koreksi atau dapat juga berfungsi sebagai nilai *threshold* (nilai ambang batas). Impuls pada sistem syaraf manusia dimodelkan sebagai nilai yang dimasukkan melalui jalur nilai masukan ke dalam model neuron. Nilai pada setiap jalur masukan akan dikalikan dengan bobot jalur, dan kemudian dijumlahkan. Nilai yang dihasilkan dari penjumlahan seluruh masukan tersebut dapat berada pada *range* yang sangat besar sehingga akan mempersulit pengolahan selanjutnya. Untuk menghindari hal tersebut, nilai hasil penjumlahan akan diproses terlebih dulu dengan menggunakan sebuah fungsi ϕ

(phi) yang disebut fungsi aktivasi. Penghitungan nilai hasil dapat dituliskan secara matematis sebagai berikut:

$$y_k = \varphi(u)$$

$$y_k = \varphi\left(\sum_{i=0}^m x_i w_{ki}\right)$$

Proses pembelajaran pada suatu neuron dilakukan dengan memperbarui nilai bobot setiap masukan (w_i) sesuai dengan pembelajaran yang diberikan. Bentuk data pembelajaran adalah berupa pasangan vektor nilai masukan ke neuron dan nilai target keluaran (t). Nilai masukan akan dimasukkan ke dalam neuron, dan menghasilkan nilai keluaran (y). Kesalahan dari nilai keluaran yang dihasilkan dihitung, dan nilai bobot w untuk setiap masukan diperbarui menggunakan:

$$w_i(n+1) = w_i(n) + \Delta w_i$$

$$\Delta w_i = \eta(t - y)x_i$$

Dengan η (mu) adalah kecepatan pembelajaran, yaitu seberapa besar bobot w terbaru pada setiap langkah pembelajaran. Proses ini dilakukan berulang-ulang dengan data pembelajaran yang berbeda. Pada akhirnya, bobot nilai masukan akan berisi nilai bobot yang sesuai.

Perancangan dan Implementasi Sistem

Secara garis besar, aplikasi pengenalan sinyal ucapan ini terdiri dari dua bagian, yaitu bagian pengenalan ucapan dan pengiriman data ke robot. Sistem ini diwujudkan ke dalam suatu perangkat lunak (program) menggunakan bahasa pemrograman Matlab 6.5. Secara umum pembuatan program ini mengikuti alur sesuai yang ditunjukkan pada Gambar



Diagram alir program utama

Pembuatan sistem pengenalan ucapan terdiri dari proses pengelolaan basis data, proses ekstraksi ciri, dan proses pemodelan dan pelatihan. Ketiga proses tersebut sering disebut dengan proses pembelajaran. Proses pengenalan ucapan merupakan proses perbandingan antara suara uji dengan model suara yang didapat dari proses pembelajaran.

Sistem yang dibuat ditujukan untuk dapat mengenali beberapa kata masukan, yaitu "maju", "mundur", "kanan", "kiri", dan "stop". Masing-masing kata mewakili satu perintah masukan pada robot. Kata "maju" untuk menggerakkan robot maju ke depan, "mundur" untuk memutar posisi robot dengan sudut 180° dan menggerakkan ke depan, "kanan" untuk menggerakkan robot belok ke kanan, "kiri" untuk menggerakkan robot belok ke kiri, dan kata "stop" untuk menghentikan robot.

Pengelolaan Basis Data

Dalam pembuatan basis data dilakukan perekaman suara dengan 6 orang responden yang terdiri dari 5 orang pria dan 1 orang wanita dengan rentang usia 22-23 tahun. Tiap responden melakukan perekaman sebanyak 10 kali untuk setiap kata, sehingga diperoleh 50 suara basis data untuk tiap responden. *Database* ditambah dengan perekaman suara di luar kata "maju", "mundur", "kanan", "kiri", dan "stop" yang akan dikenali sebagai "Suara Tidak Dikenali" sebanyak 30 suara, sehingga jumlah total keseluruhan basis data ialah 330 suara. Perekaman suara dalam pembuatan basis data ini dilakukan selama 1 detik untuk tiap ucapan. Pencuplikan dilakukan pada frekuensi 8000 Hz dengan resolusi 16 bit.

Kecepatan pencuplikan tersebut dilakukan dengan didasarkan asumsi bahwa sinyal percakapan (*speech*) berada pada daerah frekuensi 300-3400 Hz sehingga memenuhi kriteria Nyquist yang menyatakan bahwa frekuensi penyampling minimal 2 kali frekuensi maksimal dari sinyal yang akan disampling untuk menghindari *aliasing* :

$$f_s \geq 2f_h \quad f_h = f_{m \text{ tertinggi}}$$

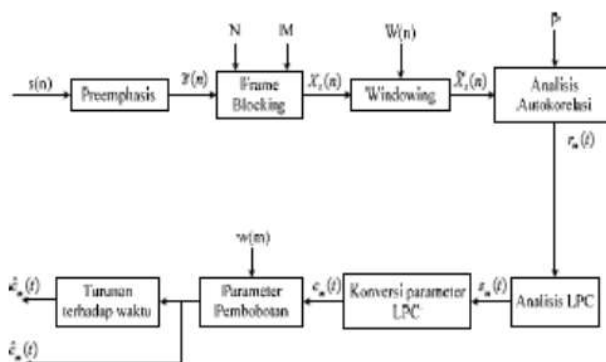
Ekstraksi Ciri

Salah satu metode yang digunakan untuk proses ekstraksi ciri adalah *Linear Predictive Coding* (LPC). Prinsip dasar dari pemodelan sinyal dengan menggunakan LPC adalah bahwa contoh sinyal ucapan $s(n)$ pada waktu ke- n dapat diperkirakan sebagai

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \dots + a_p s(n-p)$$

kombinasi linier p sampel sinyal ucapan sebelumnya yaitu

dengan koefisien $a_1, a_2, \dots, a_p$ diasumsikan bernilai konstan pada frame analisis ucapan. Prosedur untuk mendapatkan koefisien LPC diperlihatkan pada gambar di bawah ini:



Blok diagram analisis LPC

Algoritma Transformasi Fourier Cepat (Fast Fourier Transform)

Untuk lebih memudahkan dalam perhitungan, transformasi Fourier diskret selanjutnya dapat dikembangkan dengan cara mengefisienkan perulangan perhitungan juga sifat simetri dari faktor fasa WN yang menjadi sifat transformasi Fourier diskret dengan tujuan mengurangi

jumlah perhitungan yang dibutuhkan. Untuk lebih memudahkan dalam manipulasi atau perhitungan penjumlahan,

$$x(t) = \frac{ao}{2} + \sum_{k=1}^{\infty} (a_k \cos(\omega kt) + b_k \sin(\omega kt))$$

dapat dituliskan kembali dalam bentuk bilangan kompleks eksponensial berdasarkan identitas Euler

$$\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2}, \sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

Sehingga,

$$\begin{aligned} x(t) &= ao + \sum_{k=1}^{\infty} (a_k \frac{e^{j\omega kt} + e^{-j\omega kt}}{2} + b_k \frac{e^{j\omega kt} - e^{-j\omega kt}}{2j}) \\ &= ao + \sum_{k=1}^{\infty} (\frac{1}{2}(a_k - b_k)e^{j\omega kt} + \frac{1}{2}(a_k + b_k)e^{-j\omega kt}) \end{aligned}$$

dinyatakan $y_n = 1/2 (a_n - j b_n)$, dapat kita gunakan fakta-fakta bahwa a_n dan b_n adalah koefisien Fourier fungsi periodik, dengan fungsi kosinus adalah fungsi periodik genap, dan fungsi sinus adalah fungsi periodik ganjil.

Algoritma FFT (Fast Fourier Transform) yang banyak digunakan secara luas adalah algoritma dengan bilangan dasar 2, artinya masukan yang dibutuhkan sebanyak $N = 2^V$. Hal yang pertama dilakukan dalam pengembangan algoritma FFT (Fast Fourier Transform) dengan bilangan dasar 2 ialah dengan menyederhanakan penulisan persamaan DFT (Discrete Fourier Transform) dan juga membuat beberapa hubungan matematis, sehingga persamaan (2.19) dapat ditulis kembali menjadi :

$$X_1(k) = \sum_{n=0}^{N-1} x_n W_N^{kn}, k = 0, 1, \dots, N-1$$

Dan juga faktor e

$-j2\pi/N$ akan ditulis sebagai W_N , maka

$$W_N = e^{-j2\pi/N}$$

Sehingga persamaan (2.27) menjadi :

$$X_1(k) = \sum_{n=0}^{N-1} x_n W_N^{kn}, k = 0, 1, \dots, N-1$$

Akan sangat bermanfaat pada saat ini untuk memperhatikan hubungan yang melibatkan W_N pertama

$$W_N^2 = (e^{-j2\pi/N})^2 = e^{-j2\pi \cdot 2/N} = e^{-j2\pi(N/2)} = W_{N/2}$$

Kedua

$$\begin{aligned} W_N^{(K+N/2)} &= W_N^K W_N^{N/2} = W_N^K e^{-j(2\pi/N)(N/2)} \\ &= W_N^K e^{-j\pi} (\cos \pi - j \sin \pi) \\ &= -W_N^K \end{aligned}$$

bila hasilnya disimpulkan yang berhubungan dengan W_N maka kita dapatkan

$$W_N = e^{-j2\pi/N}$$

$$W_N^2 = W_{N/2}$$

$$W_N^{(K+N/2)} = -W_N^K$$

Prosedur ini dijelaskan sebagai berikut: sufiks n, dalam persamaan (2.28) terletak dari $n = 0$ hingga $n = N-1$, berhubungan dengan nilai data $X_0, X_1, X_2, X_3, \dots, X_{N-1}$. Runtun nomor genap adalah $X_0, X_2, X_4, X_6, \dots, X_{N-2}$ dan runtun bernomor ganjil adalah $X_1, X_3, X_5, \dots, X_{N-1}$, kedua runtun mengandung $N/2$ poin. Istilah dalam runtun genap dapat diwakili dengan X_{2n} dengan $n = 0$ hingga $n = N/2 - 1$ sementara untuk runtun ganjilnya menjadi X_{2n+1} , maka persamaan (2.28) dapat ditulis ulang menjadi :

$$\begin{aligned} x_1(k) &= \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nK} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{(2n+1)K} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nK} + W_N^K \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{2nK}, k = 0, 1, \dots, N-1 \end{aligned}$$

dengan menggunakan

$$W_N^{2nK} = W_{N/2}^{nK}$$

sehingga persamaan menjadi :

$$x(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_{N/2}^{nK} + W_N^K \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_{N/2}^{nK}, k = 0, 1, \dots, N-1$$

persamaan dapat ditulis menjadi :

$$X_1(k) = X_{11}(k) + W_N^K X_{12}(k), k = 0, 1, \dots, N-1$$

Bila dibandingkan persamaan (2.28) dan persamaan (2.34) terlihat bahwa $X_{11}(k)$ sesungguhnya adalah transformasi Fourier diskret runtun genap, sementara $X_{12}(k)$ merupakan runtun ganjil. Karenanya seperti yang sudah disebutkan bahwa transformasi Fourier diskret : $X_{11}(k)$ dan $X_{12}(k)$ faktor

$$W_{N/2}^K$$

terdapat di kedua $X_{11}(k)$ dan $X_{12}(k)$ dan hanya perlu dihitung sekali saja.

Proses pengestraksian ciri merupakan proses untuk mendapatkan parameter-parameter sinyal suara. Parameter-parameter inilah yang nantinya digunakan untuk membedakan satu kata dengan kata yang lain. Proses awal ekstraksi ciri yaitu dengan normalisasi amplitudo terlebih dahulu, setelah itu dilakukan pendeteksian awal dan akhir kata. Normalisasi amplitudo dilakukan dengan cara membagi semua nilai input dengan nilai maksimum dari input itu sendiri. Sehingga untuk semua sinyal masukan memiliki nilai maksimum yang sama yaitu 1.

Sedangkan deteksi awal-akhir digunakan pada proses untuk mendeteksi mulai sinyal ucapan awal dan berakhir ketika sudah tidak diucapkan. Sehingga tidak disalah artikan untuk tiap sinyal yang masuk. Setelah itu sinyal terlebih dahulu disamakan jumlah datanya, sehingga pada proses analisa LPC bisa didapat jumlah frame dan panjang masing-masing frame yang sama. Walaupun tahap ini akan memperpanjang data dan proses pada jaringan syaraf tiruan, tahap ini mampu memperkecil kesalahan pengenalan kata. Terutama untuk kata

yang memiliki jumlah suku kata berbeda. Proses normalisasi amplitudo, pendeteksian awal dan akhir kata, serta penyamaan panjang data pada Matlab dilakukan dengan fungsi 'clean' sebagai berikut :

```
[majul,fs] = wavread('mahmudmajul.wav');  
potong_majul = clean(majul);
```

Data dari hasil normalisasi selanjutnya melewati proses *preemphasis*. Setelah melalui proses *preemphasis*, sinyal suara dibagi dalam beberapa *frame* sepanjang *N sample* suara dengan jarak antara *frame* yang berdekatan dipisahkan oleh *M sample*. Setiap *frame* kemudian dijabarkan menggunakan *Hamming window*, dianalisis autokorelasi, dan dianalisis LPC sehingga didapatkan koefisien LPC. Semua proses tersebut dapat dilakukan dengan satu fungsi 'lpc_cep'.

```
koef_lpc1 = lpc_cep(potong_majul);
```

```
koef_lpc1 = rerata(koef_lpc1);
```

Proses *frame blocking* yang dilakukan pada sistem ini ditetapkan tiap 30 mili detik dengan jarak antar *frame* 10 mili detik. Jadi dengan kecepatan cuplik sebesar 8000 Hz maka tiap *frame* akan berisi 240 *byte* data dengan jarak antar *frame* 80 *byte* data atau dengan kata lain *overlap* yang terbentuk sebesar 160 *byte* data. Dengan ketentuan *frame* seperti di atas, maka untuk data hasil cuplik sebanyak 5000 data maka akan terbentuk $60 \cdot 80 \cdot 5000 \cdot 160 = -$ *frame* (hasil pembulatan dari 60,5).

Untuk perhitungan koefisien LPC, digunakan orde LPC 12 sehingga diperoleh data output sebanyak $(12+1) \times 60 = 780$ data. Selanjutnya koefisien LPC ini di rata-rata sehingga diperoleh hasil akhir koefisien LPC sebanyak 13×1 data.

Selanjutnya koefisien LPC tersebut diproses dengan *Fast Fourier Transform* (FFT) untuk mendapatkan sinyal pada domain frekuensi. Hal ini bertujuan agar perbedaan antar pola kata yang satu dengan yang lain terlihat lebih jelas sehingga ekstraksi parameter sinyal memberikan hasil yang lebih baik. Contoh proses FFT ditunjukkan oleh senarai program berikut :

```
fft_majul = abs(fft(koef_lpc1,512));
```

Proses *Fast Fourier Transform* (FFT) yang digunakan memakai 32, 64, 128 dan 512 *point* dan karena hasil FFT simetris maka keluaran FFT tersebut hanya diambil setengahnya saja. Dari data FFT ini kemudian dijadikan masukan bagi jaringan syaraf tiruan.

Perhitungan Transformasi Fourier Diskrit (DFT)

Tujuan perhitungan ini adalah untuk menyajikan solusi bagaimana proses DFT dengan menggunakan jaringan syaraf tiruan. Untuk melatih jaringan saraf dengan jumlah "input data" - "data target"-pasang waktu sinyal dan menghubungkan dengan spektrum frekuensi dan jaringan yang diubah dengan cara yang memadai oleh algoritma. Uji coba yang dilakukan dengan arsitektur jaringan yang berbeda, nomor yang berbeda, lapisan dan neuron dalam lapisan ini dengan fungsi transfer yang berbeda menunjukkan bahwa dengan upaya realistis, struktur jaringan neural bisa saja tidak belajar untuk memproses perhitungan DFT untuk sinyal input tak terlihat. Strategi harus diterapkan untuk memetakan rumus matematika DFT pada struktur jaringan syaraf tiruan. Konsep yang disajikan dalam berikut: dengan

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-jkn \frac{2\pi}{N}} \quad (1)$$

menggunakan DFT, garis spektrum dari suatu sinyal waktu diskrit dihitung dengan rumus Bagian ini sekarang menyelidiki bagaimana jaringan saraf dapat digunakan untuk menghitung DFT dari waktu sinyal dengan menggunakan formula ini. Vektor masukan dari jaringan saraf adalah waktu sinyal $x(n)$ pada titik-titik N diskrit waktu dan output nilai X

spektral (k). Untuk mendapatkan nilai spektral X (k), masing-masing waktu N nilai x (n) harus dikalikan dengan faktor:

$$e^{-jkn\frac{2\pi}{N}}$$

Ternyata tugas ini dapat dilakukan oleh neuron tanpa bias, dengan fungsi transfer linier, dan dengan nilai sebagai bobot dari sinyal input. Untuk menghitung keseluruhan spektrum, untuk setiap neuron diperlukan nilai frekuensi. 20 neuron paralel yang diperlukan, masing-masing menerima nilai-nilai input yang sama dan hanya berbeda jika mereka tergantung pada bobot nilai k. Untuk melihat gambaran umumnya, nilai-nilai bobot dari hubungan tertentu tidak digambarkan. Seperti contoh dari faktor berikut:

$$e^{-jkn\frac{2\pi}{N}}$$

formula 1 akan berbeda satu sama lain karena nilai-nilai n dan k. Semua koneksi yang berasal dari input x (0) selalu memiliki berat

$$e^{-jk\cdot 0\frac{2\pi}{N}} = 1$$

Koneksi yang berasal dari input x (1) memiliki nilai yang tergantung pada k output. Untuk sisa koneksi, prinsip yang sama untuk menghitung bobot dapat diterapkan dengan mengubah nilai dari n.

$$\begin{aligned} &[e^{-j0\cdot 1\frac{2\pi}{N}}, e^{-j1\cdot 1\frac{2\pi}{N}}, \dots, e^{-j(k-1)\cdot 1\frac{2\pi}{N}}, e^{-jk\cdot 1\frac{2\pi}{N}}] = \\ &[1, e^{-j\frac{2\pi}{N}}, \dots, e^{-j(k-1)\frac{2\pi}{N}}, e^{-jk\frac{2\pi}{N}}] \end{aligned}$$

Daftar pustaka

- Irfandy, Mahmud.2004. *Aplikasi Pengenalan Ucapan Dengan Jaringan Syaraf Tiruan Propagasi Balik Untuk Pengendalian Robot Bergerak*. Semarang: Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro.
- Velik,Rosemarie.2006.*Discrete Fourier Transform Computation Using Neural Networks*.Vienna:Vienna University of Technology.